



成都众山科技有限公司产品说明书

产品型号：ZS-AI8-D08

全部资料下载地址：<http://ask.zstel.com:8090>

技术支持服务电话：028-64267900

技术支持专员企业 QQ：3183329475

官网网站：<https://www.zstel.com/>

硬件/软件技术定制热线：19150158475 张工

目录

目录.....	2
一、产品概述.....	3
1.1 概述.....	3
1.2 性能特点.....	3
1.3 技术参数.....	3
二、外观尺寸.....	4
2.1 产品外观.....	4
2.2 产品尺寸图.....	4
三、产品接线图、跳线、指示灯说明.....	5
3.1 接线图.....	5
3.2 跳线.....	6
3.3 LED 指示灯	7
四、软件操作.....	8
4.1 配置软件.....	8
4.2 配置基本参数.....	9
4.3 DO 输出相关参数.....	9
4.4 AI 模拟量采集相关参数.....	10
4.5 AI 模拟通道参数校准参数（数值转换）	11
4.6 短信告警内容参数配置（设备需包含有短信模块）	13
4.7 其他功能（精度校准）	14
五、ModbusRTU 通讯协议、组态软件软件说明.....	16
5.1 通讯协议.....	16
5.2 寄存器地址.....	16
5.2 Modbus RTU 功能码.....	17
5.3 Modbus 通讯实例	17
六、协议详解.....	19
6.1 功能码描述.....	19
6.2 错误码描述.....	25
6.3 CRC 校验算法.....	25
七、附录.....	27
附录一：	27
附录二：	29
附录三：	30

一、产品概述

1.1 概述

ZS-AI8-D08 是一款工业级标准模拟量采集产品，共有 8 个测量通道。每个通道均可以分别设置多种 量程；RS-485 通讯接口使用标准 Modbus RTU 协议，符合工业标准。

1.2 性能特点

- 防死机硬件看门狗
- 5~35V 带防反接、过压过流保护电源
- 8 路模拟量电流输入 0~20mA
- 8 路模拟量电压输入 0~10V、0~30V
- 8 路 NPN 开关量信号输出
- 12 位分辨率，0.1%精度 ADC
- 高性能低功耗 32 位 ARM 嵌入式 CPU
- 支持 ModbusRTU 从站协议
- 12 路指示灯
- 带防雷、静电保护 RS485 通讯接口
- 工业温度范围，应对严苛现场环境
- 自定义线性模拟量数据转换

1.3 技术参数

模拟量接口	AI	8 路单端
	AI 分辨率	12bit
	AI 量程	0~10V、0~30V、0~20mA
	精度	0.1%
	采集速度	1KHz
	AI 输入阻抗	0~20mA ≤120 Ω 0~10V/0~30V ≥ 10K Ω
开关量接口	D0	8 路单端
	输出信号类型	NPN
	输出电流电压	Vout=VCC 最大负载 3W
通讯接口	通讯接口	RS485
	波特率	1200~115200bps
	数据格式	N、E、0.8.1
	通讯协议	ModbusRTU
	过压过流保护	45V
电源参数	电源规格	DC 5~35V
	功耗	12V-25mA
工作环境	工作温度、湿度	-40℃~85℃，0%RH~95%RH

其他	尺寸	82*50*32
----	----	----------

二、外观尺寸

2.1 产品外观



2.2 产品尺寸图

见文档“[第七章 附录一：外形尺寸](#)”

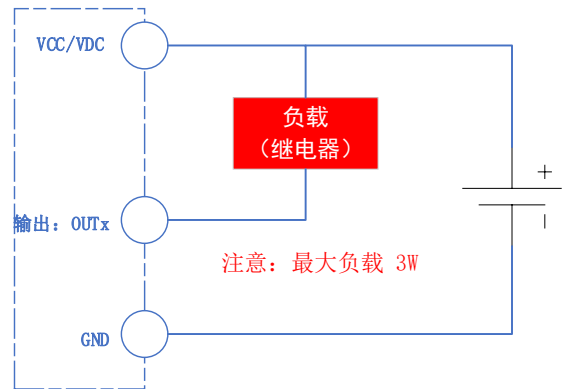
三、产品接线图、跳线、指示灯说明

3.1 接线图



顶部 8 槽接线位：

OUT1: D01 输出端 OUT3: D03 输出端 OUT5: D05 输出端 OUT7: D07 输出端
OUT2: D02 输出端 OUT4: D04 输出端 OUT6: D06 输出端 OUT8: D08 输出端

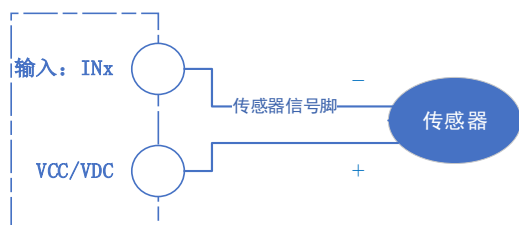


注：数字输出接继电器示意图：

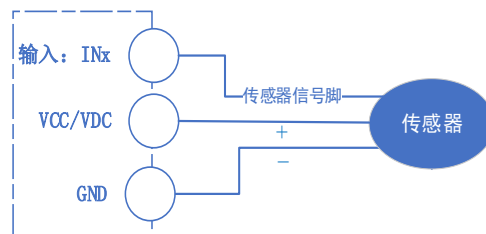
底部 12 槽接线位：

VDC: 电源正极端	AI1: 模拟量输入端 1	AI5: 模拟量输入端 5
GND: 电源负极端	AI2: 模拟量输入端 2	AI6: 模拟量输入端 6
A+: 485 通讯线 A	AI3: 模拟量输入端 3	AI7: 模拟量输入端 7
B-: 485 通讯线 B	AI4: 模拟量输入端 4	AI8: 模拟量输入端 8

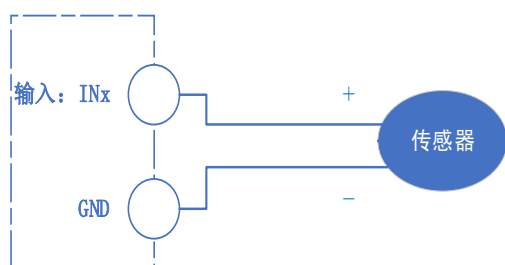
两线制传感器（无源）：



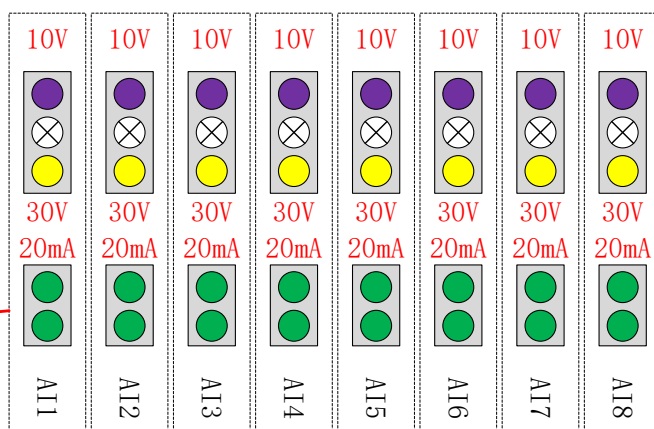
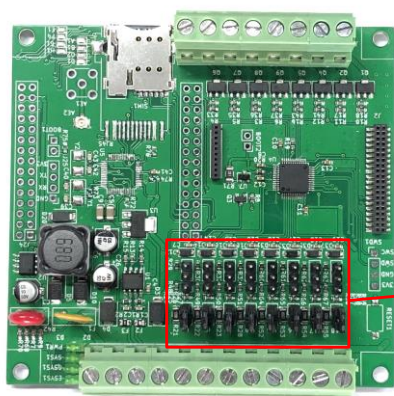
三线制传感器（有源）：



有源传感器：



3.2 跳线

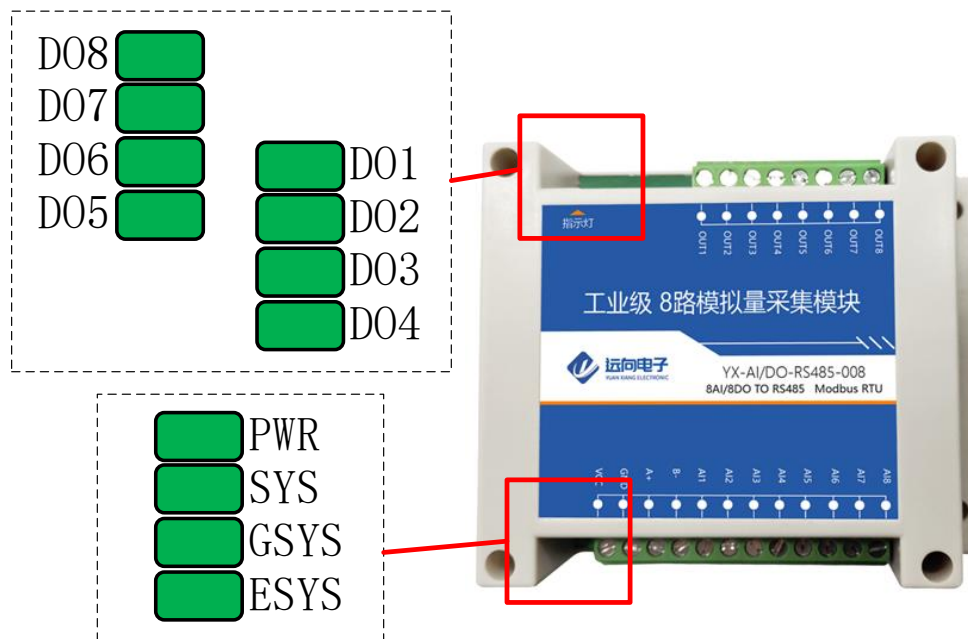


电路板上 8 个跳线座，通过跳线选择不同点量程，结合配置软件实现多量程功能：

● 通道 AIx：

- 0~10V 量程：选择紫色侧 2 个跳线座
- 0~30V 量程：选择黄色侧 2 个跳线座
- 0~20mA 量程：选择绿色侧 1 个跳线座

3.3 LED 指示灯



12 个 LED 指示灯：

顶部 8 个指示灯

- **D1:** D01 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D2:** D02 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D3:** D03 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D4:** D04 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D5:** D05 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D6:** D06 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D7:** D07 输出状态指示灯。亮：D0 有输出 灭：D0 无输出
- **D8:** D08 输出状态指示灯。亮：D0 有输出 灭：D0 无输出

底部 4 个指示灯

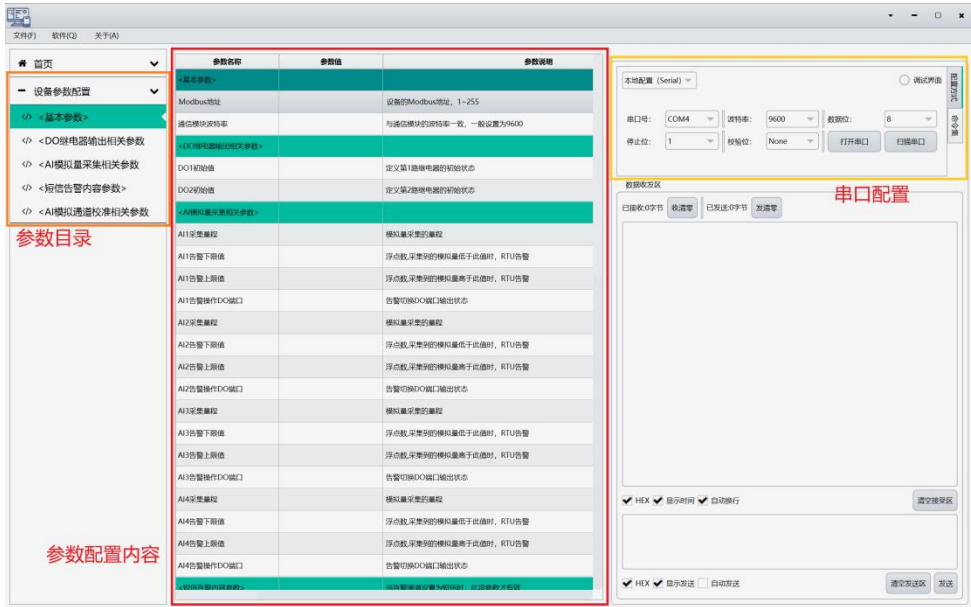
- **ESYS:** 预留
- **GSYS:** 预留
- **PWR:** 正常供电时常亮
- **SYS:** 系统状态灯，正常运行时每秒闪烁一次

四、软件操作

设备参数配置教程，结合《用户测试文档》即可对设备进行简单测试

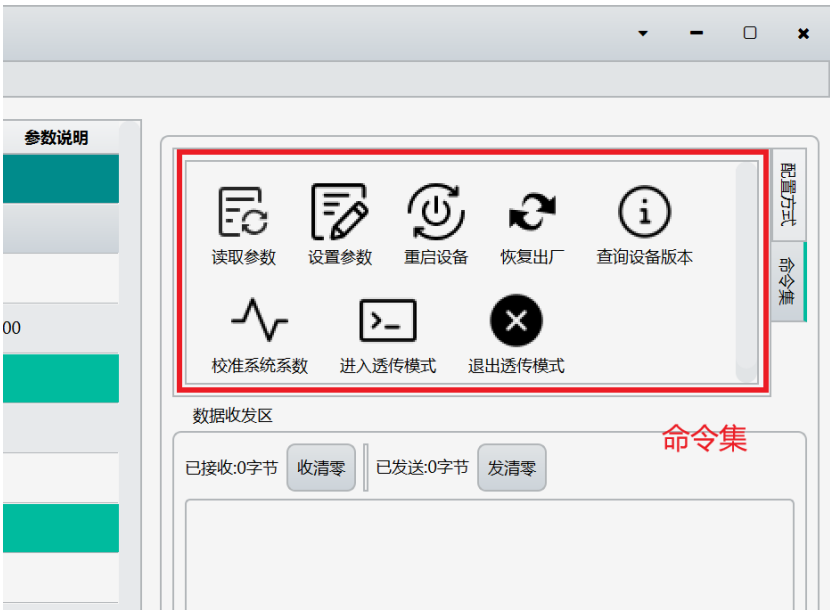
4.1 配置软件

参数配置软件介绍：



参数配置准备：

- (1) 用 USB-485 工具连接设备到电脑
- (2) 在串口配置框内配置串口波特率、停止位、校验位、数据位；（默认波特率 9600，数据位 8，停止位 1，校验位 None）
- (3) 选择串口配置框子项“命令集”



- (4) 点击“读取参数”命令按钮，读取设备参数（不同设备拥有不同指令集）
- (5) 双击对应参数项的“参数值”栏即可对参数进行修改

4.2 配置基本参数

- **Modbus 地址**：Modbus 地址参数
- **通讯模块波特率**：设备 485 通讯波特率（波特率支持主流的波特率选项）

参数名称	参数值	参数说明
<基本参数>		
Modbus地址		设备的Modbus地址, 1~255
通信模块波特率		与通信模块的波特率一致, 一般设置为9600
<DO继电器输出相关参数>		
DO1初始值		定义第1路继电器的初始状态
DO2初始值		定义第2路继电器的初始状态
<AI模拟量采集相关参数>		
AI1采集量程		
AI1告警下限值		告警
AI1告警上限值		告警
AI1告警操作DO端口		
AI2采集量程		
AI2告警下限值		
AI2告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI2告警操作DO端口		告警切换DO端口输出状态

通信模块波特率

1200

1200

2400

4800

9600

19200

38400

57600

115200

取消

4.3 DO 输出相关参数

- **DOx 初始值**：双击“参数值”栏，通过选择参数可配置开机上电后 DO 输出状态

<DO继电器输出相关参数>		
DO1初始值		定义第1路继电器的初始状态
DO2初始值		定义第2路继电器的初始状态
DO3初始值		
DO4初始值		
DO5初始值		
DO6初始值		
DO7初始值		
DO8初始值		
<AI模拟量采集相关参数>		

DO1初始值

低 (断开)

低 (断开)

高 (导通)

记忆上一次的状态

确定

取消

4.4 AI 模拟量采集相关参数

- **AIx 采集量程**：双击“参数值”栏，选择对应选项修改 AIx 采集量程

<AI模拟量采集相关参数>		
AI1采集量程	双击配置参数	模拟量采集的量程
AI1告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI1告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI1告警操作DO端口		告警切换DO端口输出状态
AI2采集量程		
AI2告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI2告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI2告警操作DO端口		
AI3采集量程		
AI3告警下限值		
AI3告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI3告警操作DO端口		告警切换DO端口输出状态
AI4采集量程		模拟量采集的量程
AI4告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI4告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI4告警操作DO端口		告警切换DO端口输出状态

AI1采集量程

电压 0 ~ 10V

电压 0 ~ 10V

电压 0 ~ 30V

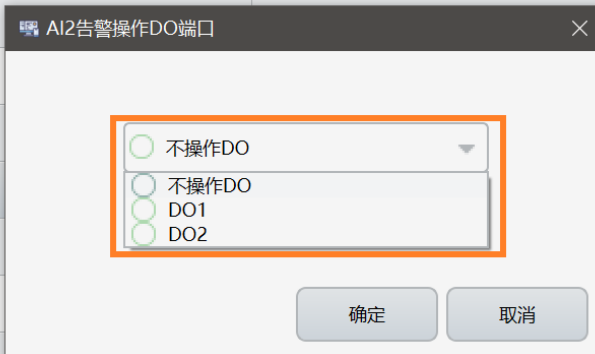
电流 0 ~ 20mA

确定

取消

- **AIx 告警下限值**：采集数据低于此值是执行 DO 操作（“AIx 告警操作 DO 端口”中配置具体操作端口项），通过**模块**发送“安装地址”+“设备身份 ID”+“AIx 低于下限告警短信内容”（具体内容参考“**告警内容案例**”）
- **AIx 告警上限值**：采集数据高于此值是执行 DO 操作（“AIx 告警操作 DO 端口”中配置具体操作端口项），通过**模块**发送“安装地址”+“设备身份 ID”+“AIx 高于上限告警短信内容”（具体内容参考“**告警内容案例**”）
- **AIx 告警操作 DO 端口**：双击“参数值”栏，选择对应选项修改 AIx 告警操作端口

参数名称	参数值	参数说明
<DO继电器输出相关参数>		
DO1初始值		定义第1路继电器的初始状态
DO2初始值		定义第2路继电器的初始状态
<AI模拟量采集相关参数>		
AI1采集量程		模拟量采集的量程
AI1告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI1告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI1告警操作DO端口	双击配置参数	告警切换DO端口输出状态
AI2采集量程		
AI2告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI2告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI2告警操作DO端口		告警切换DO端口输出状态
AI3采集量程		
AI3告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI3告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI3告警操作DO端口		告警切换DO端口输出状态
AI4采集量程		模拟量采集的量程
AI4告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI4告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI4告警操作DO端口		告警切换DO端口输出状态



4.5 AI 模拟通道参数校准参数（数值转换）

此参数用于转换模拟量值（例：将电压转换为温度值）
转换满足线性公式：

$$AI = (x + gain) \times ratio + offset$$

AI 即转换后数值：

x 为转换前数值

gain 为增益系数（初始为 0.0）

ratio 为比例系数（初始为 1.0）

offset 为位置系数（初始为 0.0）

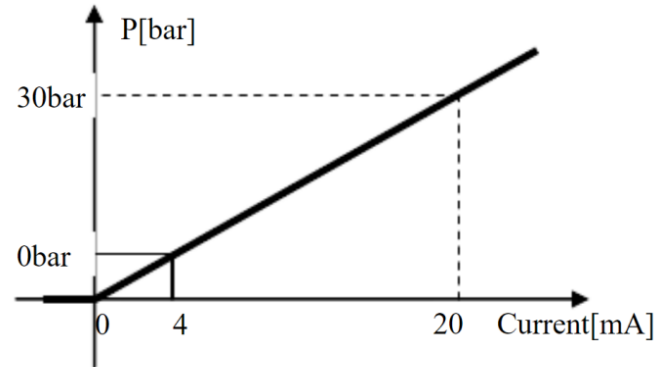
<AI模拟通道校准相关参数>		
AI1增益值 gain		AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI1比例值 ratio		AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI1位移值 offset		AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI2增益值 gain		AI2采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI2比例值 ratio		AI2采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI2位移值 offset		AI2采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI3增益值 gain		AI3采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI3比例值 ratio		AI3采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI3位移值 offset		AI3采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI4增益值 gain		AI4采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI4比例值 ratio		AI4采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI4位移值 offset		AI4采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$

配置案例：

一款压力传感器输出线性曲线如下，连接传感器输出到设备“CH1”通道。

4~20mA 压力传感器

电流越大，压力越大。
压力测量范围 0~30bar



$$\begin{aligned} \text{ratio} &= 30 / 20 = 1.5 \\ \text{gain} &= 0 \\ \text{offset} &= 0 \end{aligned}$$

根据以上计算原理计算出“ratio: 1.5”“gain: 0”“offset: 0”，将计算得出的参数值通过配置软件设置到设备中。

设备此时采样通道值读数值即为读取转换后的通道数据 $\text{value} = (AIx + 0) * 1.5 - 0$

<AI模拟通道校准相关参数>		
AI1增益值 gain	0	AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI1比例值 ratio	1.5	AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI1位移值 offset	0	AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI2增益值 gain		AI2采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI2比例值 ratio		AI2采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI2位移值 offset		AI2采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI3增益值 gain		AI3采集的模拟量计算公式 $AI = (adc + gain) * ratio$
AI3比例值 ratio		AI3采集的模拟量计算公式 $AI = (adc + gain) * ratio$

配置完参数后重启设备，给定传感器压力，再通过 modbus 03 功能码读取 CH1 通道值“9.12”即为实际压强值

4.6 短信告警内容参数配置（设备需包含有短信模块）

注：需要配合本公司支持 4G 网络的短信远程模块

- 安装地址：主要用于报警通知使用（设备需包含有短信模块）
- 设备身份 ID：主要用于报警通知使用（设备需包含有短信模块）
- AIx 低于下限告警短信内容：AIx 告警内容
- AIx 恢复短信内容：AIx 告警内容
- AIx 超出上限告警短信内容：AIx 告警内容

<短信告警内容参数>		当告警通道设置为短信时，此项参数才有效
安装地址		最长16个汉字,32个字符，短信报警用
设备身份ID		8位编码，短信报警时有效
AI1告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI1低于下限告警短信内容		最多30个汉字或60个字符
AI1恢复短信内容		最多30个汉字或60个字符
AI1超出上限告警短信内容		最多30个汉字或60个字符
AI2告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI2低于下限告警短信内容		最多30个汉字或60个字符
AI2恢复短信内容		最多30个汉字或60个字符
AI2超出上限告警短信内容		最多30个汉字或60个字符
AI3告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI3低于下限告警短信内容		最多30个汉字或60个字符
AI3恢复短信内容		最多30个汉字或60个字符
AI3超出上限告警短信内容		最多30个汉字或60个字符
AI4告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI4低于下限告警短信内容		最多30个汉字或60个字符
AI4恢复短信内容		最多30个汉字或60个字符
AI4超出上限告警短信内容		最多30个汉字或60个字符

告警内容案例：

配置如下：

安装地址：“北厂区：”

设备身份 ID：“05 号设备：”

AI1 低于下限值短信内容：“低温警告”

<短信告警内容参数>		当告警通道设置为短信时，此项参数才有效
安装地址	北厂区：	最长16个汉字,32个字符，短信报警用
设备身份ID	05号设备：	8位编码，短信报警时有效
AI1告警周期	1	循环告警的时间间隔，单位为分钟，0表示只告警一次
AI1低于下限告警短信内容	低温告警	最多30个汉字或60个字符
AI1恢复短信内容	恢复正常水平	最多30个汉字或60个字符
AI1超出上限告警短信内容	高温告警	最多30个汉字或60个字符
AI2告警周期	1	循环告警的时间间隔，单位为分钟，0表示只告警一次
AI2低于下限告警短信内容	低压强告警	最多30个汉字或60个字符
AI2恢复短信内容	恢复正常	最多30个汉字或60个字符
AI2超出上限告警短信内容	超压强告警	最多30个汉字或60个字符
AI3告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI3低于下限告警短信内容		最多30个汉字或60个字符
AI3恢复短信内容		最多30个汉字或60个字符
AI3超出上限告警短信内容		最多30个汉字或60个字符
AI4告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI4低于下限告警短信内容		最多30个汉字或60个字符
AI4恢复短信内容		最多30个汉字或60个字符
AI4超出上限告警短信内容		最多30个汉字或60个字符

AI 低于下限告警短信：“北厂区：05 号设备：低温警告”

其他告警短信内容格式与低于下限值格式一致（消息内容为空则不发送内容）

4.7 其他功能（精度校准）

4.7.1 校准模拟量精度

当系统精度不够时，产品提供一个用户精度校准功能

（1）设备串口连接上电脑，点击“命令集”中“校准模拟量精度”按钮进行模拟量校准



(2) 取消勾选显示区“HEX”复选框，等到接受区提示如下内容时执行下一步

数据收发区

已接收:4819字节 收清零 已发送:189字节 发清零

【14:20:24.693】下行:CH1: 10
【成功】
【14:20:31.317】上行:■U
【14:20:36.473】上行:
校准系数: 1.0000, 1.0002
串口等待接受校准模拟量数值
串口发送格式如下:
CH1: xx
CH2: xx
【14:20:38.078】下行:CH1: 10
【14:20:38.238】上行:模拟通道1校准完成
【14:20:38.400】上行:模拟通道1系数: 0.9986
【14:20:38.597】上行:退出模拟量校准模式
【重启设备】
【14:20:45.546】下行:AA550004E00600EA
【成功】
【14:20:45.719】上行:■U
【成功】
【14:20:46.325】上行:HIGH

☐ HEX ☒ 显示时间 ☒ 自动换行 清空接收区

CH1: 10

☐ HEX ☒ 显示发送 ☐ 自动发送 清空发送区 发送

(3) 给定模拟量 AIx 通道恒定基准电压（注意通道量程）

(4) 发送通道基准电压格式 “CHx: xx”（例：“CH1: 10”）注意取消勾选发送区“HEX”复选按钮

(5) 等待设备提示校准消息

数据收发区

已接收:4819字节 收清零 已发送:189字节 发清零

【14:20:24.693】下行:CH1: 10
【成功】
【14:20:31.317】上行:■U
【14:20:36.473】上行:
校准系数: 1.0000, 1.0002
串口等待接受校准模拟量数值
串口发送格式如下:
CH1: xx
CH2: xx
【14:20:38.078】下行:CH1: 10
【14:20:38.238】上行:模拟通道1校准完成
【14:20:38.400】上行:模拟通道1系数: 0.9986
【14:20:38.597】上行:退出模拟量校准模式
【重启设备】
【14:20:45.546】下行:AA550004E00600EA
【成功】
【14:20:45.719】上行:■U
【成功】
【14:20:46.325】上行:HIGH

☐ HEX ☒ 显示时间 ☒ 自动换行 清空接收区

CH1: 10

☐ HEX ☒ 显示发送 ☐ 自动发送 清空发送区 发送

(6) 重启设备

五、ModbusRTU 通讯协议、组态软件软件说明

5.1 通讯协议

本产品兼容标准 Modbus RTU 从站协议，能够支持标准 Modbus RTU 组态软件，详细介绍参考“[第六章 Modbus 协议详解](#)”

快速上手可参照“[5.3 Modbus 通讯实例](#)”

5.2 寄存器地址

寄存器地址	名称	字节数	说明	备注
模拟量输入				
0x0000 (0)	AI1_H	2	模拟量通道 1 高	每个模拟量通道占 2 个 Modbus 寄存器，4 个字节，格式为浮点数，浮点 数格式符合 IEEE 754 标准 可参照 5.3 读取 AI
0x0001 (1)	AI1_L	2	模拟量通道 1 低	
0x0002 (2)	AI2_H	2	模拟量通道 2 高	
0x0003 (3)	AI2_L	2	模拟量通道 2 低	
0x0004 (4)	AI3_H	2	模拟量通道 3 高	
0x0005 (5)	AI3_L	2	模拟量通道 3 低	
0x0006 (6)	AI4_H	2	模拟量通道 4 高	
0x0007 (7)	AI4_L	2	模拟量通道 4 低	
0x0008 (8)	AI5_H	2	模拟量通道 5 高	
0x0009 (9)	AI5_L	2	模拟量通道 5 低	
0x000A (10)	AI6_H	2	模拟量通道 6 高	
0x000B (11)	AI6_L	2	模拟量通道 6 低	
0x000C (12)	AI7_H	2	模拟量通道 7 高	
0x000D (13)	AI7_L	2	模拟量通道 7 低	
0x000E (14)	AI8_H	2	模拟量通道 8 高	
0x000F (15)	AI8_L	2	模拟量通道 8 低	
0x0100 (256)	AI1_D	2	模拟量通道 1	AI 整数寄存器 0~20mA: 000~2000 0~10V: 000~1000 0~30V: 000~3000
0x0101 (257)	AI2_D	2	模拟量通道 2	
0x0100 (258)	AI3_D	2	模拟量通道 1	
0x0101 (259)	AI4_D	2	模拟量通道 2	
0x0100 (260)	AI5_D	2	模拟量通道 1	
0x0101 (261)	AI6_D	2	模拟量通道 2	
0x0100 (262)	AI7_D	2	模拟量通道 1	
0x0101 (263)	AI8_D	2	模拟量通道 2	
数字量输出				
00014	D01	2	数字量通道 1	0000 表示断开 0001 表示闭合
00015	D02	2	数字量通道 2	
00016	D03	2	数字量通道 3	
00017	D04	2	数字量通道 4	
00018	D05	2	数字量通道 5	
00019	D06	2	数字量通道 6	

0001A	D07	2	数字量通道 7
0001B	D08	2	数字量通道 8

5.2 Modbus RTU 功能码

功能码	操作	说明
01	读取单位 DO 状态	Bit 位表示 DO 输出状态
03	读取 AI, DO 寄存器值	读取 AI, DO 寄存器值
04	读取 AI, DO 寄存器值	读取 AI, DO 寄存器值
05	写单个 DO	0xFF00: 闭合;0x0000: 断开
06	写单个 DO	0x0001: 闭合;0x0000: 断开
0F	写多个 DO	参照“ 第六章 Modbus 协议详解 ”
10	写多个 DO	参照“ 第六章 Modbus 协议详解 ”

详细讲解参照“[第六章 Modbus 协议详解](#)”

5.3 Modbus 通讯实例

(1) 读取 DO1:

a. 用 01 功能码读取 DO1

发送: 01 01 00 14 00 01 BD CE

接受: 01 01 01 00 51 88

b. 用 03 功能码读取 DO1

发送: 01 03 00 14 00 01 C4 0E

接受: 01 03 02 00 00 B8 44

c. 用 04 功能码读取 DO1

发送: 01 04 00 14 00 01 71 CE

接受: 01 04 02 00 00 B9 30

(2) 操作 DO1:

a. 用 05 功能码操作单个 DO1

发送: 01 05 00 14 FF 00 CC 3E

接受: 01 05 00 14 FF 00 CC 3E

b. 用 06 功能码操作单个 DO1

发送: 01 06 00 14 00 01 08 0E

接受: 01 06 00 14 00 01 08 0E

c. 用 0F 功能码操作多个 DO1、DO2

发送: 01 0F 00 14 00 02 01 03 AE 95

接受: 01 0F 00 14 00 02 94 0E

d. 用 10 功能码操作多个 DO1、DO2

发送: 01 10 00 14 00 02 04 00 01 00 01 63 50
接受: 01 10 00 14 00 02 01 CC

(3) 读取 AI:
给定输入 4.96 (40 9E E7 CF)

a. 用 03 功能码读取浮点数 AI1:

发送: 01 03 00 00 00 02 C4 0B

接受: 01 03 04 40 9E E7 CF 85 B9

4.96 IEE 浮点数十六进制为 (40 9E E7 CF)

b. 用 04 功能码读取浮点数 AI1:

发送: 01 04 00 00 00 02 71 CB

接受: 01 04 04 40 9E CE 1F 9A 02

C. 用 03 功能码读取整数 AI1:

发送: 01 03 01 00 00 01 85 F6

接收: 01 03 02 01 F0 B9 90

整数读出数值为 496 (0x01F0)

d. 用 04 功能码读取整数 AI1:

发送: 01 04 01 00 00 01 30 36

接收: 01 04 02 01 F0 B8 E4

整数读出数值为 496 (0x01F0)

六、协议详解

地址域	功能码	数据	差错检验
-----	-----	----	------

Modbus 使用“big-Endian”（大端模式）表示地址和数据项，这就意味着当发射多个字节时，首先发送最高字节。

例如：寄存器地址为 0x0014，首先发送的是 0x00，然后才是 0x14。

一个正常的 Modbus 响应：响应功能码=请求功能码。

一个 Modbus 的异常响应：响应功能码=请求功能码+0x80，提供一个异常码来指示差错原因。

6.1 功能码描述

6.1.1 01 读线圈

可以使用此功能码读取继电器 D01~D08 的状态。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-8 为 0-7。

响应 PDU 中 N 个字节的线圈状态的每一个 bit 位代表一个线圈的状态，状态 1=ON，0=OFF。第一个字节的最低位 LSB 代表第 0 号线圈的状态（即起始地址指定的线圈号为 0 号线圈），其他线圈依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各线圈状态。

如果线圈数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x01
起始地址	2 个字节	0x0014 至 0x001B
线圈数量	2 个字节	n(1 至 8)
CRC 校验	2 个字节	

注：线圈状态的字节数 $N = \text{线圈数量 } n / 8$ ，如果余数不等于 0，则 $N = n / 8 + 1$

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x81 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量 D01-D08 的实例

请求		响应	
地址	64	地址	64
功能码	01	功能码	01
起始地址高 H	00	字节数	01
起始地址低 L	14	DO1-DO8 状态	5A

线圈数量高 H	00	CRC 校验高 H	CF
线圈数量低 L	08	CRC 校验低 L	7F
CRC 校验高 H	74		
CRC 校验低 L	3D		

发送：640100140008743D

DTU 响应：64010101CF7F

D01-D08 的状态字节为 5A，二进制 01011010，D01 是这个字节的 LSB(第 0 位)为 0 表示断开，D02 是第 1 位为 1 表示闭合，D03 是第 2 位为 0 表示断开，D04 是第 3 位为 1 表示闭合，D05 是第 4 位为 1 表示闭合，D06 是第 5 位为 0 表示断开，D07 是第 6 位为 1 表示闭合，D08 是第 7 位为 0 表示断开。

6.1.2 03 读保持寄存器

6.1.3 04 读输入寄存器

使用该功能码可以读取所有寄存器包括 AI1-AI8、D01-D08 的状态。

请求 PDU 详细说明了起始寄存器地址和寄存器数量，从零开始寻址寄存器，因此寻址寄存器 1-32 为 0-32。

响应报文中的寄存器数据每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2。对于 AI1-AI8，一个通道占用 2 个寄存器，4 个字节的值使用浮点数表示，对于 D01-D08，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 04
起始地址	2 个字节	0x0000 至 0x001B
寄存器数量	2 个字节	n(1 至 32)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 0x04
字节数	1 个字节	N=2*n
寄存器值	N 个字节	N=2*n, n 为寄存器数量
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x83 或 0x84 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读模拟量输入 AI1-AI8 的

请求		响应	
地址	64	地址	64
功能码	03	功能码	03
起始地址高 H	00	字节数	30
起始地址低 L	00	AI1-AI8 值	16 个字节
寄存器数量高 H	00	CRC 校验高 H	
寄存器数量低 L	10	CRC 校验低 L	
CRC 校验高 H	4C		
CRC 校验低 L	35		

发送：6403000000103433

6.1.4 05 写单个线圈

可以使用该功能码写单个继电器 D01-D08 为断开或闭合

请求数据域中的常量说明请求的 ON/OFF 状态，十六进制值 0xFF00 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)，其他所有值都是非法的，对输出不起作用，DTU 返回错误响应。

请求域中的输出地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x001B
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x001B
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x85 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 D02 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64

功能码	05	功能码	05
输出地址高 H	00	输出地址高 H	00
输出地址低 L	15	输出地址低 L	15
输出值高 H	FF	输出值高 H	FF
输出值低 L	00	输出值低 L	00
CRC 校验高 H	94	CRC 校验高 H	94
CRC 校验低 L	0B	CRC 校验低 L	0B

发送：64050015FF00940B

DTU 响应：64050015FF00940B

6.1.5 06 写单个寄存器

可以使用该功能码写单个继电器 D01-D08 为断开或闭合。

请求数据域中的寄存器值说明请求的 ON/OFF 状态，十六进制值 0001 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)。

请求域中的寄存器地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x001B
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x001B
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x86 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 D02 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64
功能码	06	功能码	06
寄存器地址高 H	00	寄存器地址高 H	00
寄存器地址低 L	15	寄存器地址低 L	15
寄存器值高 H	00	寄存器值高 H	00

寄存器值低 L	01	寄存器值低 L	01
CRC 校验高 H	50	CRC 校验高 H	50
CRC 校验低 L	3B	CRC 校验低 L	3B

发送：640600150001503B

DTU 响应：640600150001503B

6.1.6 0F 写多个线圈

可以使用此功能码写多个继电器 D01~D08 为断开或闭合。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-8 为 0-7。

请求数据域中的内容说明了被请求的 ON/OFF 状态，域比特位中的逻辑“1”请求相应输出为 ON，域比特位中的逻辑“0”请求相应输出为 OFF。从数据域中第一个字节的 bit0 开始到 bit7，然后到第二个字节的 bit0，依次表示第一个线圈到第 n 个线圈的 ON/OFF 值。

正常响应返回功能码、起始地址和线圈数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x001B
线圈数量	2 个字节	n(1 至 8)
字节数	1 个字节	$N=n/8$, 或 $N=n/8+1$
输出值	N 个字节	
CRC 校验	2 个字节	

注：线圈输出字节数 $N=$ 线圈数量 $n/8$ ，如果余数不等于 0，则 $N=n/8+1$

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x001B
线圈数量	2 个字节	n(1 至 8)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x8F (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求从线圈 D01 开始写入 4 个线圈的实例

请求		响应	
地址	64	地址	64
功能码	0F	功能码	0F

起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
线圈数量高 H	00	线圈数量高 H	00
线圈数量低 L	02	线圈数量低 L	02
字节数	01	CRC 校验高 H	9D
输出值	02	CRC 校验低 L	FB
CRC 校验高 H	68		
CRC 校验低 L	82		

发送：640F001400020103A942

DTU 响应：640F001400029DFB

D01-D02 的输出值为 02，二进制 00000010，D01 是这个字节的 LSB(第 0 位) 为 0 表示断开，D02 是第 1 位为 1 表示闭合，用 0 填充剩余未使用的 6 位。

6.1.7 10 写多个寄存器

使用该功能码可以写连续寄存器 D01-D08 的状态。

请求 PDU 详细说明了起始寄存器地址、寄存器数量、字节数和寄存器值，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

寄存器数据中每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

正常响应返回功能码、起始地址和被写入寄存器的数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x001B
寄存器数量	2 个字节	n(1 至 8)
字节数	1 个字节	$N=2*n$
寄存器值	N 个字节	$N=2*n$, n 为寄存器数量
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x001B
寄存器数量	2 个字节	n(1 至 8)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x90 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个控制继电器 D01-D04 的实例

请求		响应	
地址	64	地址	64
功能码	10	功能码	10
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
寄存器数量高 H	00	寄存器数量高 H	00
寄存器数量低 L	02	寄存器数量低 L	02
字节数	04	CRC 校验高 H	08
DO1 寄存器值高 H	00	CRC 校验低 L	39
DO1 寄存器值高 L	01		
DO2 寄存器值高 H	00		
DO2 寄存器值高 L	00		
DO3 寄存器值高 H	4D		
DO3 寄存器值高 L	5D		
DO4 寄存器值高 H	64		
DO4 寄存器值高 L	10		
CRC 校验高 H	00		
CRC 校验低 L	14		

发送：64100014000204000100004D5D

DTU 响应：6410001400020839

D01 寄存器值为 0001 表示闭合，D02 寄存器值为 0000 表示断开

6.2 错误码描述

错误码含义：当 DTU 收到错误的 Modbus 指令时，会返回功能码为请求功能码+0x80，紧随着一个字节的错误码代表出错原因。

错误码 01：表示不支持的功能码，众山 DTU 支持上述 8 种功能码，除此之外的功能码都会返回错误码为 01 的错误。

错误码 02：表示起始地址不存在或者起始地址加上寄存器数量后的地址不存在。总的来说表示访问的寄存器不存在。

错误码 03：表示寄存器数量不符合规范或者寄存器值非法。

错误码 04：表示读写寄存器错误。

6.3 CRC 校验算法

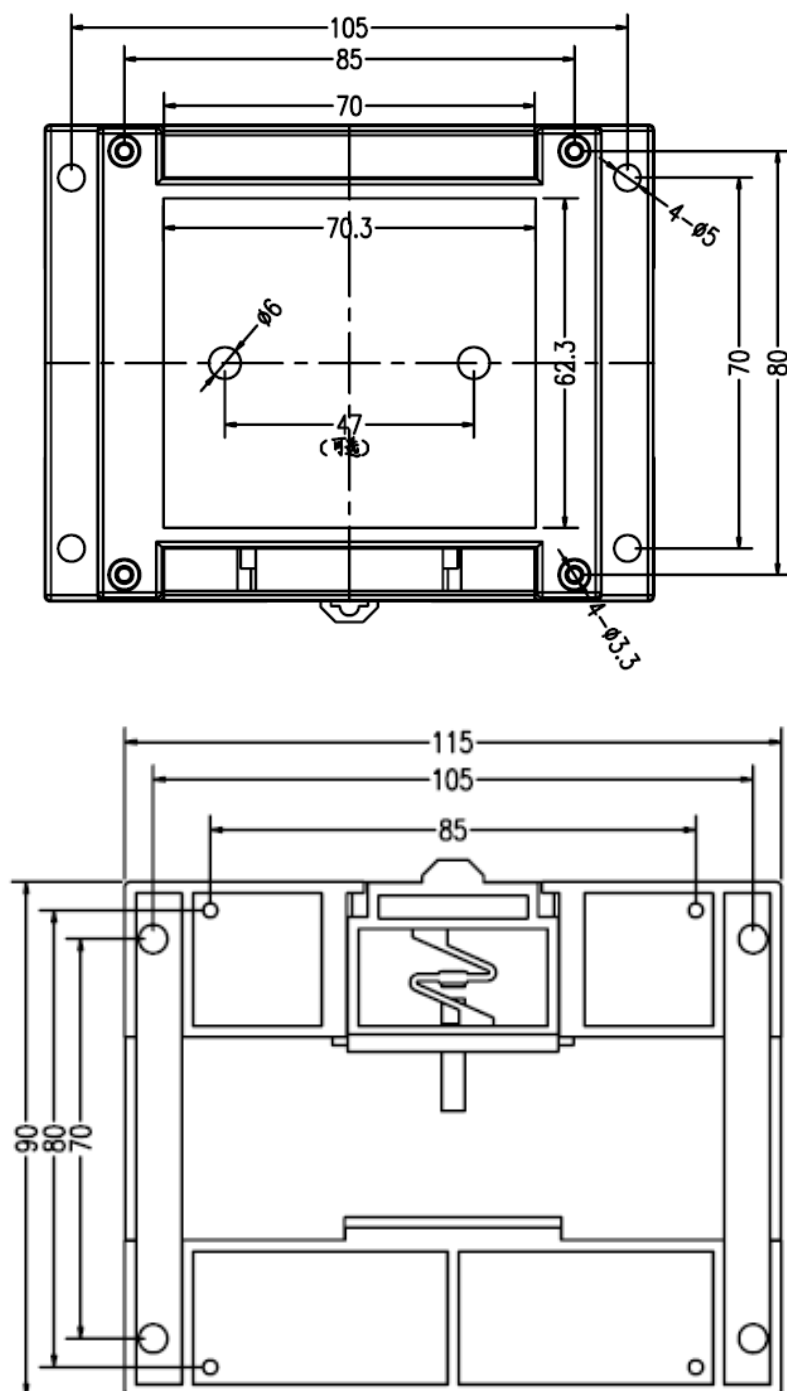
CRC 即[循环冗余校验码](#)（Cyclic Redundancy Check）：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

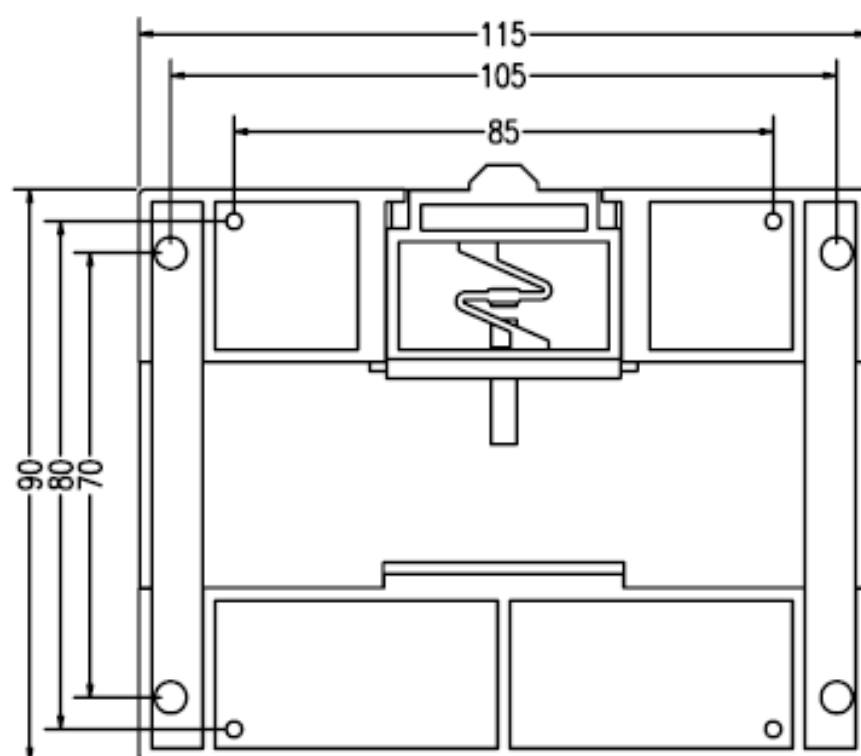
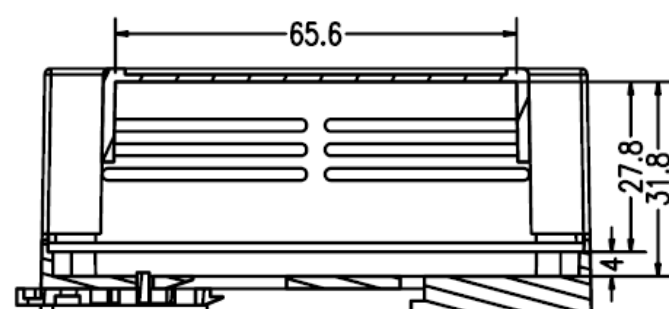
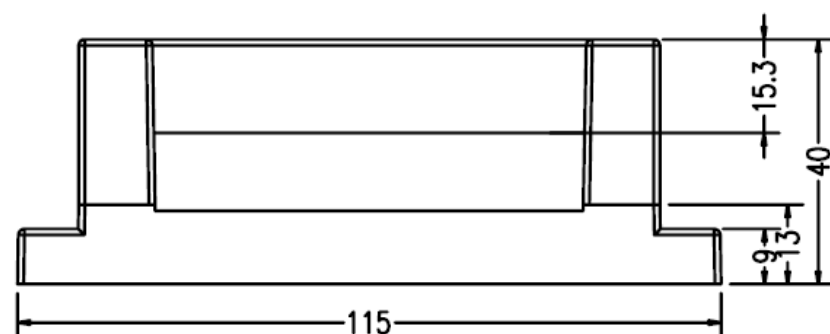
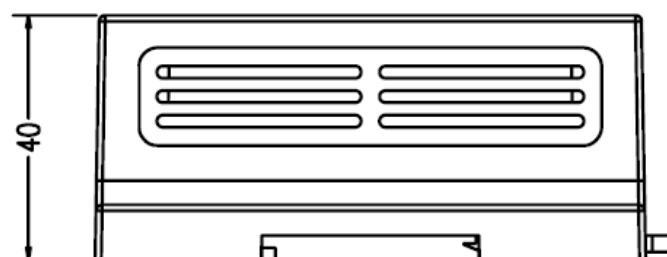
详细计算方法见“[第七章 附录二：CRC 校验算法](#)”
附 CRC 校验算法代码：

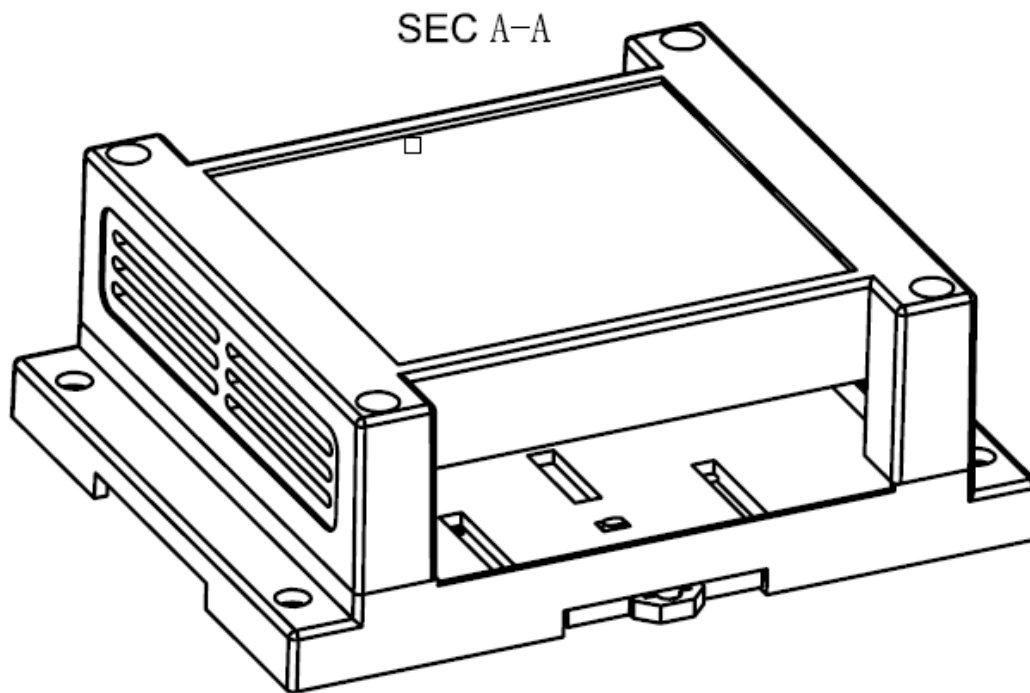
七、附录

附录一：

外形尺寸







附录二：

CRC 校验算法：

```
uint16_t mb_crc( uint8_t *snd, uint16_t num )
{
    uint8_t CRC_Lb, CRC_Hb;
    uint8_t CRC_L, CRC_H;
    uint16_t crc;

    CRC_H = 0xFF;
    CRC_L = 0xFF;

    for ( uint16_t i = 0; i < num; i++ ) {
        CRC_L = CRC_L ^ snd[i];
        for ( uint16_t j = 0; j < 8; j++ ) {
            CRC_Lb = CRC_L;
            if ((CRC_L & 1) == 1) {
                CRC_L = (CRC_L - 1) / 2;
                CRC_Lb = CRC_L;
                CRC_Hb = CRC_H;
            }
            if ( (CRC_H & 1) == 1 ) {
                CRC_L = CRC_L + 128;
                CRC_Lb = CRC_L;
                CRC_H = (CRC_H - 1) / 2;
                CRC_Hb = CRC_H;
            }
        }
    }
}
```

```

    } else {
        CRC_H = CRC_H / 2;
        CRC_Hb = CRC_H;
    }
    CRC_L = CRC_L ^ 1;
    CRC_Lb = CRC_L;
    CRC_H = CRC_H ^ 0xA0;
    CRC_Hb = CRC_H;
}
else
{
    CRC_L = CRC_L / 2;
    CRC_Lb = CRC_L;
    CRC_Hb = CRC_H;
    if ( (CRC_H & 1) == 1 ) {
        CRC_L = CRC_L + 128;
        CRC_Lb = CRC_L;
        CRC_H = (CRC_H - 1) / 2;
        CRC_Hb = CRC_H;
    } else {
        CRC_H = CRC_H / 2;
        CRC_Hb = CRC_H;
    }
}
}
}
crc = CRC_L;
crc <<= 8;
crc |= CRC_H;
return crc;
}

```

附录三：

更新记录

v1.3

修改

- * 参数表格 电源 5~40 改为 5~35V
- * 尺寸说明放置在附录一
- * CRC 校验算法放置在附录二

新增

- * NPN 开关量输出功能特性描述
- * Modbus 示例高亮指示
- * 表格行色差

- * 文档内部快捷跳转链接

v1.2

新增

- * 1.3 技术参数 DO 开关量参数

v1.1

新增

- * 第七章，modbus 协议详解内容

修改

- * 初稿错别字表述

v1.0

编写

- * 4 路模拟量产品说明书初稿